UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/551,891	08/28/2006	Martin Vorbach	2885/96	3388

26646 7590 04/05/2010
KENYON & KENYON LLP
ONE BROADWAY
NEW YORK, NY 10004

EXAMINER
VICARY, KEITH E

ART UNIT	PAPER NUMBER
2183	

MAIL DATE	DELIVERY MODE
04/05/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	**Application No.** 10/551,891	**Applicant(s)** VORBACH, MARTIN	
	Examiner KEITH VICARY	**Art Unit** 2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *16 March 2010*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *5-11* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *5-11* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date *See Continuation Sheet*.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

Continuation of Attachment(s) 3). Information Disclosure Statement(s) (PTO/SB/08), Paper No(s)/Mail Date :3/16/2010 (first), 3/16/2010 (second).

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

0. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 3/16/2010 has been entered.

1. Claims 5-11 are pending in this office action and presented for examination. Claims 5-10 have been newly amended by amendment filed 3/16/2010.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 5 and 8-10 are rejected under 35 U.S.C. 103(a) as being unpatentable over Smith et al. (Smith) (US 6658564) in view of Borkenhagen et al. (Borkenhagen) (US 6076157) in view of Jones et al. (Jones) (US 5812844).

4. Consider claim 5, Smith discloses processing in accordance with a first configuration (e.g. col. 11, lines 60-61, compiling hardware functions into configuration

patterns; col. 2, lines 29-31 disclose of configuration data being used to execute an application) having a maximum allowed runtime (col. 8, line 66 to col. 9, line 4; switching between different functions at predetermined time intervals; in other words, the length of the predetermined time interval is the maximum allowed runtime); and if no interrupt occurs, reconfiguring the reconfigurable unit with a second configuration in response to expiry of the maximum allowed runtime (col. 8, line 66 to col. 9, line 4; switching between different functions at predetermined time intervals; in other words, a different configuration will be switched to once the predetermined time length of the previous configuration expires), the maximum allowed runtime expiring due to suppression by at least one of a task switch and a thread switch of an increase of the maximum allowed runtime (col. 8, line 66 to col. 9, line 4; switching between different functions at predetermined time intervals; the switching between functions at predetermined time intervals keeps a function from executing for longer than its predetermined time length).

However, Smith does not disclose of increasing, by the first configuration, the first configuration's maximum allowed runtime. Smith also does not explicitly disclose of if an interrupt occurs, suppressing the increase in response to the interrupt.

On the other hand, Borkenhagen discloses of increasing, by the first thread, the first thread's maximum allowed runtime (col. 15, lines 1-19, disclose of not forcing a thread switch if no other thread is ready to process instructions).

Thread switches entail latency and performance penalties (Borkenhagen, col. 12, lines 23-24).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Borkenhagen with the invention of Smith in order to lessen latency and performance penalties. Note that Borkenhagen's teaching regarding threads, as applied to Smith's invention of using configurations to implement functions, teaches of the overall limitation of increasing, by the first configuration, the first configuration's maximum allowed runtime.

However, although Smith and Borkenhagen disclose of suppressing an increase in maximum allowed runtime (Smith via the function switching of col. 8, line 66 to col. 9, line 4 and Borkenhagen in the scenario in which there are other threads ready to process instructions), Smith and Borkenhagen do not explicitly disclose of if an interrupt occurs, suppressing the increase in response to the interrupt.

On the other hand, Jones discloses of threads for performing device interrupt handling which are scheduled against tasks performed by other executing programs (col. 5, lines 26-29).

Jones' teaching prevents time-critical threads from being suspended for significant periods of time (Jones, col. 2, lines 33-35). Additionally, the non-preemptive nature of Jones' interrupt handling decreases the amount of thread switches which, would lessen latency and performance penalties (e.g. see Borkenhagen, col. 12, lines 23-24).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Jones with the invention of Smith and Borkenhagen in order to prevent time-critical threads from being suspended for significant periods of time and lessen latency and performance penalties. Note that Jones' teaching of threads for performing device interrupt handling, when implemented into the invention of Smith and Borkenhagen wherein the existence of other functions ready to process instructions suppresses an increase in maximum allowed runtime, results in the overall limitation of, when an interrupt occurs, suppressing the increase in response to the interrupt.

5. Consider claim 8, Smith discloses processing in accordance with a configuration (e.g. col. 11, lines 60-61, compiling hardware functions into configuration patterns; col. 2, lines 29-31 disclose of configuration data being used to execute an application) having a maximum allowed runtime (col. 8, line 66 to col. 9, line 4; switching between different functions at predetermined time intervals; in other words, the length of the predetermined time interval is the maximum allowed runtime).

However, Smith does not disclose of triggering an increase, by the configuration, of the configuration's maximum allowed runtime. Smith also does not explicitly disclose of, responsive to an interrupt, suppressing an increase by the configuration of the maximum allowed runtime to respond to the interrupt upon expiry of the maximum allowed runtime.

On the other hand, Borkenhagen discloses of triggering an increase, by the thread, of the thread's maximum allowed runtime (col. 15, lines 1-19, disclose of not forcing a thread switch if no other thread is ready to process instructions).

Thread switches entail latency and performance penalties (Borkenhagen, col. 12, lines 23-24).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Borkenhagen with the invention of Smith in order to lessen latency and performance penalties. Note that Borkenhagen's teaching regarding threads, as applied to Smith's invention of using configurations to implement functions, teaches of the overall limitation of increasing, by the first configuration, the first configuration's maximum allowed runtime.

However, although Smith and Borkenhagen disclose of suppressing an increase in maximum allowed runtime (Smith via the function switching of col. 8, line 66 to col. 9, line 4 and Borkenhagen in the scenario in which there are other threads ready to process instructions), Smith and Borkenhagen do not explicitly disclose of, responsive to an interrupt, suppressing an increase by the configuration of the maximum allowed runtime to respond to the interrupt upon expiry of the maximum allowed runtime.

On the other hand, Jones discloses of threads for performing device interrupt handling which are scheduled against tasks performed by other executing programs (col. 5, lines 26-29).

Jones' teaching prevents time-critical threads from being suspended for significant periods of time (Jones, col. 2, lines 33-35). Additionally, the non-preemptive nature of Jones' interrupt handling decreases the amount of thread switches which, would lessen latency and performance penalties (e.g. see Borkenhagen, col. 12, lines 23-24).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Jones with the invention of Smith and Borkenhagen in order to prevent time-critical threads from being suspended for significant periods of time and lessen latency and performance penalties. Note that Jones' teaching of threads for performing device interrupt handling, when implemented into the invention of Smith and Borkenhagen of, responsive to the existence of other functions ready to process instructions, suppressing an increase by the configuration of the maximum allowed runtime to respond to the other functions upon expiry of the maximum allowed runtime, results in the overall limitation of, responsive to an interrupt, suppressing an increase by the configuration of the maximum allowed runtime to respond to the interrupt upon expiry of the maximum allowed runtime.

6. Consider claim 9, Smith discloses reconfiguring the reconfigurable unit (col. 2, line 9, reconfigurable computer system) with a new configuration (e.g. col. 11, lines 60-61, compiling hardware functions into configuration patterns; col. 2, lines 29-31 disclose of configuration data being used to execute an application) for handling a function responsive to expiry of the maximum allowed runtime (col. 8, line 66 to col. 9, line 4; switching between different functions at predetermined time intervals; in other words, the length of the predetermined time interval is the maximum allowed runtime).

However, Smith does not disclose of increasing, by a configuration having a maximum allowed runtime, the configuration's maximum allowed runtime. Smith also does not disclose of suppressing the increase in response to an interrupt, and reconfiguring the reconfigurable unit with a new configuration for handling the interrupt responsive to expiry of the maximum allowed runtime.

On the other hand, Borkenhagen discloses of increasing, by a thread having a maximum allowed runtime, the thread's maximum allowed runtime (col. 15, lines 1-19, disclose of not forcing a thread switch if no other thread is ready to process instructions).

Thread switches entail latency and performance penalties (Borkenhagen, col. 12, lines 23-24).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Borkenhagen with the invention of Smith in order to lessen latency and performance penalties. Note that Borkenhagen's teaching regarding threads, as applied to Smith's invention of using configurations to implement functions, teaches of the overall limitation of increasing, by a configuration having a maximum allowed runtime, the configuration's maximum allowed runtime.

However, although Smith and Borkenhagen disclose of suppressing an increase in maximum allowed runtime (Smith via the function switching of col. 8, line 66 to col. 9, line 4 and Borkenhagen in the scenario in which there are other threads ready to process instructions), Smith and Borkenhagen do not explicitly disclose of suppressing the increase in response to an interrupt, and reconfiguring the reconfigurable unit with a new configuration for handling the interrupt responsive to expiry of the maximum allowed runtime.

On the other hand, Jones discloses of threads for performing device interrupt handling which are scheduled against tasks performed by other executing programs (col. 5, lines 26-29).

Jones' teaching prevents time-critical threads from being suspended for significant periods of time (Jones, col. 2, lines 33-35). Additionally, the non-preemptive nature of Jones' interrupt handling decreases the amount of thread switches which, would lessen latency and performance penalties (e.g. see Borkenhagen, col. 12, lines 23-24).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Jones with the invention of Smith and Borkenhagen in order to prevent time-critical threads from being suspended for significant periods of time and lessen latency and performance penalties. Note that Jones' teaching of threads for performing device interrupt handling, when implemented into the invention of Smith and Borkenhagen of suppressing the increase in response to other functions ready to process instructions, and reconfiguring the reconfigurable unit with a new configuration for handling a function responsive to expiry of the maximum allowed runtime, teaches the overall limitation of suppressing the increase in response to an interrupt, and reconfiguring the reconfigurable unit with a new configuration for handling the interrupt responsive to expiry of the maximum allowed runtime.

7. Consider claim 10, Smith discloses processing in accordance with a first configuration (e.g. col. 11, lines 60-61, compiling hardware functions into configuration

patterns; col. 2, lines 29-31 disclose of configuration data being used to execute an application) having a maximum allowed runtime (col. 8, line 66 to col. 9, line 4; switching between different functions at predetermined time intervals; in other words, the length of the predetermined time interval is the maximum allowed runtime); if an interrupt does not occur: for a scheduled task switch, the counter counting to the maximum allowed runtime without a retriggering of the counter by the first configuration (col. 8, line 66 to col. 9, line 4; switching between different functions at predetermined time intervals; in other words, the length of the predetermined time interval is the maximum allowed runtime; it is inherent that some form of counter is necessary to measure the predetermined time intervals); and responsive to the reaching of the maximum allowed runtime, performing one of a task switch and a thread switch by reconfiguring the reconfigurable unit with a second configuration (col. 8, line 66 to col. 9, line 4; switching between different functions at predetermined time intervals; in other words, a different configuration will be switched to once the predetermined time length of the previous configuration expires).

However, Smith does not disclose of the first configuration triggering a counter reset, the counter reset increasing the maximum allowed runtime. Smith also does not explicitly disclose of if an interrupt does occur, responsive to the occurrence of the interrupt, the maximum allowed runtime is not increased.

On the other hand, Borkenhagen discloses of the first thread increasing the maximum allowed runtime (col. 15, lines 1-19, disclose of not forcing a thread switch if no other thread is ready to process instructions).

Thread switches entail latency and performance penalties (Borkenhagen, col. 12, lines 23-24).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Borkenhagen with the invention of Smith in order to lessen latency and performance penalties. Note that Borkenhagen's teaching regarding threads and increasing the maximum allowed runtime, as applied to Smith's invention of using configurations to implement functions and using a counter to switch at predetermined time intervals, teaches of the overall limitation of the first configuration triggering a counter reset, the counter reset increasing the maximum allowed runtime.

However, although Smith and Borkenhagen disclose of suppressing an increase in maximum allowed runtime (Smith via the function switching of col. 8, line 66 to col. 9, line 4 and Borkenhagen in the scenario in which there are other threads ready to process instructions), Smith and Borkenhagen do not explicitly disclose of if an interrupt does occur, responsive to the occurrence of the interrupt, the maximum allowed runtime is not increased.

On the other hand, Jones discloses of threads for performing device interrupt handling which are scheduled against tasks performed by other executing programs (col. 5, lines 26-29).

Jones' teaching prevents time-critical threads from being suspended for significant periods of time (Jones, col. 2, lines 33-35). Additionally, the non-preemptive nature of Jones' interrupt handling decreases the amount of thread switches which, would lessen latency and performance penalties (e.g. see Borkenhagen, col. 12, lines 23-24).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Jones with the invention of Smith and Borkenhagen in order to prevent time-critical threads from being suspended for significant periods of time and lessen latency and performance penalties. Note that Jones' teaching of threads for performing device interrupt handling, when implemented into the invention of Smith and Borkenhagen wherein, if there are functions with instructions ready to be processed, responsive to the occurrence of the functions with instructions ready to be processed, the maximum allowed runtime is not increased, results in the overall limitation of, if an interrupt does occur, responsive to the occurrence of the interrupt, the maximum allowed runtime is not increased

8. Claim 11 is rejected under 35 U.S.C. 103(a) as being unpatentable over Smith et al. (Smith) (US 6658564) in view of Borkenhagen et al. (Borkenhagen) (US 6076157).

9. Consider claim 11, Smith discloses configurable cells (col. 2, line 9, reconfigurable computer system) configurable with a configuration (e.g. col. 11, lines 60-61, compiling hardware functions into configuration patterns; col. 2, lines 29-31 disclose of configuration data being used to execute an application) having a maximum allowed runtime (col. 8, line 66 to col. 9, line 4; switching between different functions at predetermined time intervals; in other words, the length of the predetermined time interval is the maximum allowed runtime).

However, Smith does not disclose the configuration is adapted to trigger a counter reset to increase its maximum allowed runtime conditional at least upon that an interrupt is not detected and processing is to continue without a thread switch and without a task switch.

On the other hand, Borkenhagen discloses the thread is adapted to increase its maximum allowed runtime conditional at least upon that an interrupt is not detected and processing is to continue without a thread switch and without a task switch (col. 15, lines 1-19, disclose of not forcing a thread switch if no other thread is ready to process instructions).

Thread switches entail latency and performance penalties (Borkenhagen, col. 12, lines 23-24).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Borkenhagen with the invention of Smith in order to lessen latency and performance penalties. Note that Borkenhagen's teaching regarding threads and increasing the maximum allowed runtime, as applied to Smith's invention of using configurations to implement functions and using a counter to switch at predetermined time intervals, teaches of the overall limitation that the configuration is adapted to trigger a counter reset to increase its maximum allowed runtime conditional at least upon that an interrupt is not detected and processing is to continue without a thread switch and without a task switch.

10. Claim 6 is rejected under 35 U.S.C. 103(a) as being unpatentable over Smith, Borkenhagen, and Jones as applied to claim 5 above, and further in view of Parhami (Parallel Counters for Signed Binary Signals).

11. Consider claim 6, Smith, Borkenhagen, and Jones do not disclose that the first configuration triggers a parallel counter to perform the increasing.

On the other hand, Parhami discloses of a parallel counter (section 1, second paragraph, first line, parallel counter).

Parhami's teaching of a parallel counter achieves higher speeds than regular counters (section 1, first paragraph, last two lines).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Parhami with the invention of Smith, Borkenhagen, and Jones, in order to achieve higher speeds.

12. Claim 7 is rejected under 35 U.S.C. 103(a) as being unpatentable over Smith, Borkenhagen, and Jones as applied to claim 5 above, and further in view of Rubinstein et al. (Rubinstein) (US 4959781).

13. Consider claim 7, Smith discloses of an interrupt whose processing requires handling within the maximum allowed runtime (col. 8, lines 14-15, detecting a high priority function such as a real-time interrupt handling process).

However, Smith, Borkenhagen, and Jones do not disclose that the interrupt whose processing requires handling within the maximum allowed runtime is handled on a component reserved for handling of interrupts whose processing requires handling within the maximum allowed runtime and on which the first configuration is not run.

On the other hand, Rubinstein discloses of handling interrupts on a component reserved for handling of interrupts on which the configuration is not run (col. 1, lines 24-29, all interrupts from a particular class are assigned to and handled by a particular processor; classes may all be assigned to a single processor).

Rubinstein's teaching minimizes impact on other system processing (Rubinstein, col. 1, lines 42-44).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Rubinstein with the invention of Smith, Borkenhagen, and Jones in order to minimize impact on other system processing.

Response to Arguments

14. Applicant newly argues in the paragraph spanning pages 7 and 8 that nothing in the Borkenhagen reference suggests that the interrupt itself causes the inability of a thread to perform useful processing, and it is not Applicant's obligation to determine how to handle any perceived disadvantages or glitches in what is disclosed in the cited reference, namely that an interrupt may be handled once it occurs that an active thread ceases useful processing for a predetermined time period (without the interrupt itself causing such cessation by the active thread from performing the useful processing).

Examiner notes that in considering the disclosure of a reference, it is proper to take into account not only specific teachings of the reference but also the inferences which one skilled in the art would reasonably be expected to draw therefrom. Examiner generally believes that one skilled in the art would infer that the interrupt itself does cause the inability of a thread to perform useful processing. However, examiner recognizes the level of subjectivity involved regarding these inferences, and, to further prosecution, has presented a different combination of prior art which is not reliant on such implicit disclosure.

15. Applicant argues on page 8 that the citations by examiner regarding thread priority would not disclose an interrupt causing an active thread to be unable to perform useful processing or otherwise causing an active thread to suppress increasing its maximum allowed runtime, or to cease processing.

Although examiner believes that thread priority would cause an active thread to be unable to perform useful processing (e.g. the arrival of an interrupt which causes a thread's priority to be raised may result in that thread's priority to be greater than the active thread, causing a thread switch such that the active thread is no longer able to perform useful processing), examiner has not relied upon the citations regarding thread priority in the current rejection.

16. Applicant argues on pages 8-9 that Borkenhagen would not teach the newly amended limitation that expiry occurs due to the suppression by the task and/or thread switch.

In view of this newly amended limitation, examiner has introduced a new combination of prior art in order to meet the aforementioned limitation and its corresponding claim. Examiner notes that while Borkenhagen's specific teaching of "useful processing" and its possible inferences are no longer being relied upon, the Borkenhagen reference is still used to teach at least a facet of the aforementioned limitation, namely the concept of not forcing a thread switch if no other thread is ready to process instructions, as conveyed in col. 15, lines 1-19.

Conclusion

17. Any inquiry concerning this communication or earlier communications from the examiner should be directed to KEITH VICARY whose telephone number is (571)270-1314. The examiner can normally be reached on Monday - Thursday, 7:00 a.m. - 5:30 p.m., EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on 571-272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Eddie P Chan/
Supervisory Patent Examiner, Art Unit 2183

/Keith Vicary/
Examiner, Art Unit 2183